UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,467	10/28/2003	Yu Kwong Ng	2705-726	6905

20575 7590 06/19/2007

MARGER JOHNSON & MCCOLLOM, P.C.
210 SW MORRISON STREET, SUITE 400
PORTLAND, OR 97204

EXAMINER	
WALTER, CRAIG E	

ART UNIT	PAPER NUMBER
2188	

MAIL DATE	DELIVERY MODE
06/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/696,467	NG ET AL.	
	Examiner	Art Unit	
	Craig E. Walter	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *13 April 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14 and 27-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14, 27-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 13 April 2007 has been entered.

Status of Claims

2. Claims 1-14 and 27-38 are pending in the Application.

Claims 27-38 are new.

Claims 15-26 are cancelled.

Claims 1-3, 5, 7, 9 and 13 are amended.

Claims 1-14 and 27-38 are rejected.

Response to Amendment

3. Applicant's amendments and arguments filed on 13 April 2007 in response to the office action mailed on 17 January 2007 have been fully considered, but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-6 and 27-33 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 1 recites, "wherein said hash value is not stored in the memory" in lines 10-11 of the claim. The original specification does not support this newly amended limitation therefore it constitutes new matter. More specifically, Applicant describes the hash value as being used as a "pointer" to point to an address in memory. In other words, each and every hash value corresponds to a particular location within the memory storing the partial keys. Since these addresses are storage locations of the memory, the hash values are inherently stored in the memory.

A similar rejection applies to claims 27 and 36.

Claims 2-6 and 28-33 are rejected for inheriting the deficiencies of claims 1 and 27 respectively.

5. Claims 1-6, 27-33 and 36 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to

which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 1 recites, "wherein said hash value is not stored in the memory" in lines 10-11 of the claim. This phrase renders the claim non-enabling, as one of ordinary skill in the art would not be sufficiently enabled to ascertain how a generated hash value could be used without being stored in a memory, and further one of ordinary skill in the art would be unable to determine where the generated value is stored, if not in the memory in light of the original specification. In other words, the generated hash value must inherently be stored in some memory it order for it to be used in determining a memory location. Applicant's Fig 6B is an example illustrating how a hash function is used to generate a hash value. The only structural elements illustrated are the hash function (element 602) and a memory (element 601). It is clear that the hash function (element 602) is merely a function generator *per se*, incapable of storing the generated value; therefore the generated hash value must inherently be stored in the memory.

A similar rejection applies to claims 27 and 36.

Claims 2-6 and 28-33 are rejected for inheriting the deficiencies of claims 1 and 27 respectively.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1-14 and 27-38 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites, "wherein said hash value is not stored in the memory" in lines 10-11 of the claim. This phrase renders the claim indefinite, as one of ordinary skill in the art would be unable to ascertain exactly how a generated hash value could be used without being stored in a memory, and further one of ordinary skill in the art would be unable to determine where the generated value is stored, if not in the memory in light of the original specification. In other words, the generated hash value must inherently be stored in some memory it order for it to be used in determining a memory location. Applicant relies on Fig 6B as an example illustrating using a hash function to generate a hash value. The only structural elements illustrated are the hash function (element 602) and a memory (element 601). It is clear that the hash function (element 602) is merely a function generator *per se*, incapable of storing the generated value; therefore the generated hash value must inherently be stored in the memory.

A similar rejection applies to claims 27 and 36.

Claims 2-6 and 28-33 are rejected for inheriting the deficiencies of claims 1 and 27 respectively.

The phrase "a data sum of said plurality of partial keys" as recited in claim 7 (lines 4 and 5) renders the claim indefinite, as one of ordinary skill in the art would be unable to determine what exactly constitutes a "data sum". More specifically, are the

partial keys actually being summed, or does said data sum refer to the aggregate amount of memory required to store all of said partial keys? The latter is assumed.

A similar rejection applies to claims 28 and 38.

Claims 8-14 are rejected for inheriting the deficiencies of claim 7.

As for claim 30, the phrase "reading less than the original key" as recited in line 2 renders the claim indefinite. More specifically, one of ordinary skill in the art would be unable to ascertain exactly how one could "read less" than a key (i.e. is less than the total amount of data comprising the key being read here, or is other data in an amount less than the size of the original key being read?) The former is assumed.

As for claim 34, the phrase "identify the stored partial keys that is associated with the hash value" as recited in line 5 renders the claim indefinite. More specifically, Examiner presumes Applicant intended to recite, "identify the stored partial ***key*** that is associated with the hash value", however the phrase "the stored partial key that is associated with the hash value" lacks antecedent basis. For example, lines 5-7 set forth a correspondence between a stored partial key and an input key, however such a correspondence is not previously set forth between a stored partial key and the hash value as recited in this claim. Additionally, the phrase "the partial key" as recited in lines 10 and 11 lack antecedent basis as more than one partial key (i.e. "a partial key" and "a plurality of stored partial keys") are previously set forth. Which is being referenced here?

Claims 35-38 are rejected for inheriting the deficiencies of claim 34.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 7-9, 10, 34, 35, 37 and 38 are rejected under 35 U.S.C. 102(e) as being anticipated by Brandin et al., hereinafter Brandin (US Patent 6,493,813 B1).

As for claims 7 and 34, Brandin teaches an apparatus and system, comprising:

a memory which stores a plurality of partial keys used to determine hashing conflicts, wherein said plurality of partial keys correspond to a plurality of original keys, and wherein a data sum of said plurality of partial keys requires less memory than a data sum of said plurality of original keys (Fig. 13b illustrates a partial key (the 128-bit portion of depicted as elements A and B in element 324). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). Each key is split into a partial key (elements A+B) as illustrated in Fig. 13b, and subsequently provided to the transform generator – col. 2, lines 54-57. Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a partial key (i.e. Fig. 13B – each key is split into one partial key

which is comprises a majority (i.e. 128 of 196) of consecutive bits). Each partial key helps to form the constituent elements of, and correspond to, its respective original key (196-bits). The partial keys (128-bit) require less memory than the original keys (128-bits).

Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10(exp)8 keys) – The transform generator determines an address and a confirmer for each key (col. 2, lines 47-48. The information determined from each of the original keys (or partial key as shown in Fig. 13b) is used to prevent the occurrence of collisions (i.e. hashing conflicts) – col. 2 lines 21-30);

a hash function block coupled to a memory that applies any polynomial to a full key and generates a partial key and a hash value which is used to point to one of the plurality of partial keys stored in the memory wherein the plurality of partial keys include saved bits comprising a consecutive, sequential strings of bits derived from the plurality of original keys (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to the partial key in Fig 13b. – the original key (element 324) is split into a partial key, and the hash function is applied. Additionally, referring to Fig. 13b, the original key (element 324) is comprised of a partial key (elements A+B). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-49). Additionally note that addressing

and pointer information is stored directly in the memory as per col. 2, line 66 through col. 3, line 11; and

a processor that identifies keys and conflicts by comparing one of the plurality of partial keys to the partial key comprising a majority of bits of the full key (again, the partial key comprises a majority of consecutive bits of the original key (element 324, A+B), Brandin's system inherently contains a processor (whether it be either hardware, software, or a combination of the two) to perform the comparison.

As for claim 8, Brandin teaches the apparatus of Claim 7, wherein the memory comprises a 2(exp)N hash table size (referring to Fig. 3, the store table example used (element 50) contains 16 entries (i.e. N=4)).

As for claim 9, Brandin teaches the apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the memory comprises a number of bits equal to or more than the number of bits of the full key minus a number of bits of the hash value (referring again to Fig. 13B, partial key (element 324, A+B) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key (X(A) + X(AB)). Since the partial key is more than half the original key's size, the partial key is equal to or greater than the size of the original key minus the hash value – col. 7, lines 14-49).

As for claim 10, Brandin teaches the apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

As for claim 35, Brandin teaches the stored partial key as including a majority of data bits of its corresponding input key (the partial key, element 324 (A+B) is a majority subset of the full key (A+B+C), element 324).

As for claim 37, Brandin teaches the hash value includes an address location associated with the corresponding input key (the memory store as depicted in Fig. 1 stores the pointer information to locate the key).

As for claim 38, Brandin teaches a data sum of the stored partial keys which is less than a data sum of the equal number of input keys (the amount memory required to store the partial key (A+B) is less than that of the full key (A+B+C).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Rajski et al., hereinafter Rajski (US PG Publication 2002/0016806 A1).

As for claims 11 and 12, Brandin fails to teach his LFSR as corresponding to either a Fibonacci, or a Galois version.

Rajski however teaches a method for synthesizing linear finite state machines, which includes both the Fibonacci, or a Galois versions – paragraph 0002, lines 17-20 and paragraph 0003, lines 1-4 – both types are described in his teachings.

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Rajski's method for synthesizing linear finite state machines for his own LFSRs. By doing so, Brandin would be able to more efficiently implement his LFSR with fewer levels of logic, and a lower internal fan-out of the circuitry, as taught by Rajski (paragraph 0012, lines 1-18).

9. Claims 1-4, 27-30 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin in further view of Biran (US Patent 6,345,347 B1).

As for claims 1 and 27 and 36, Brandin teaches:

> storing a plurality of partial keys in memory (Fig 13b illustrates a partial key (element 324, A+B) corresponding to an equal number of original keys in memory (each original key A+B+C, contains one and only one partial key (A+B)), wherein storage of said plurality of partial keys requires less memory than storage of said equal number of original keys (128 bit partial key compared to a 196 bit original key), and wherein said plurality of partial keys are used to determine hashing conflicts (element 324). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). Each key is split into a partial key as illustrated in Fig. 13b), and are subsequently provided to the transform generator – col. 2, lines 54-57. Referring to Fig. 9, each key is stored as an entry in the

memory table (i.e. 10[exp]8 keys)). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a partial key (i.e. Fig. 13B). Each partial key (i.e. 128-bit partial key) form the constituent elements of, and correspond to, its respective original key (196-bits). The partial keys (158-bit) require less memory than the original keys (196-bits);

applying a hash function to an original key of said plurality of equal number of original keys to generate a hash value, wherein said hash function comprises any polynomial (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to the partial key in Fig 13b.);

accessing the memory according to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading a stored partial key of said plurality of partial keys from the memory corresponding to the hash value, wherein said hash value is based on said original key (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13b illustrates that this can be applied to a partial key if the original key is greater than 64 bits;

Note giving the limitation "reading a partial key from the memory that corresponds to said hash value, wherein said hash value is based on said original key" its "broadest reasonable interpretation consistent with the specification" (In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)), reading a partial key may include reading the entire key (which is split into a partial key), since the each partial key is inherently read during the step of reading the key in its entirety. The hash value will always be "based on the original key" even if said value is only made up by a portion of the original key (i.e. the portion corresponding to the partial key).

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a stored partial key stored in the memory, wherein the partial key corresponds at most with one of the stored partial keys ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). Note each 196-bit key contains 128 bits (one-one correspondence). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys). Additionally, Brandin fails to teach not storing the hash value in the memory (Brandin rather teaches storing the value in the memory).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing

the hashed values of the keys)). Additionally, Brandin teaches managing the pointer and hash value functionality in the I/O adapter (Fig. 3, element 38 – translation table), rather than in the system memory (i.e. not in the memory).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claim 28, Brandin teaches the multiple partial keys as corresponding to an equal number of multiple input keys and a data sum of all the multiple partial keys is less than a data sum of all the equal number of multiple input keys (each original key (element 324, A+B+C) contains one partial key (element 324 (A+B). The partial key is 128-bits and the original key is 196-bits).

As for claim 29, Brandin teaches multiple partial keys as each being selectable according to a different hash value derived from one of the equal number of multiple input keys (Fig. 1, the keys stored in memory are uniquely selected based on the hash value).

As for claim 30, Brandin teaches comparing of the partial key as including reading less than the original key (each partial key contains less bits than the original key so less bit are read for the comparison).

As for claim 2, Brandin teaches the partial key from the memory corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits, that is a subset of the original key where the subset includes a majority of bits of the original key (referring to Fig. 13b, the original key (element 324, A+B+C) contains one partial key (elements 324, A+B). The bits in the partial key are stored in a sequential line (based on the key length), each containing more than half of bits than the original key – col. 7, lines 14-49).

As for claim 3, Brandin teaches the stored partial key comprises a number of bits equal to or more than a number of bits of the original key minus a number of bits of the hash value (referring again to Fig. 13B, partial key A+B (element 3324) is input into the LFSR to generate a hash value (transform) which is equal or greater in size to the partial key. Since the partial key is more than half the original key's size, the partial key is equal to or great than the size of the original key minus the hash value – col. 7, lines 14-49).

As for claim 4, Brandin teaches the hash function as being is implemented by a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

10. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claim 1 above, and in further view of Ji (US PG Publication 2005/0086363 A1).

As for claim 6, Brandin teaches:

> reading a result from the memory corresponding to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49, and the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11).

Brandin fails however to teach forwarding a packet of data according to the result read from the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

11. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claim 13, Brandin fails to teach the apparatus of claim 7 further including a reverse function generator coupled to the memory wherein the reverse function generator restores the full key based on the one of the plurality of partial keys stored in the memory and the hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed, by applying the reverse hash function on the hash result and the hash identifiers for key restoration – col. 8, lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure,

which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

12. Claims 5 and 31-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claims 1 and 34 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claim 5, Brandin fails to teach applying a reverse function on the partial key from the memory corresponding to the hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21.

As for claims 32 and 33, though Brandin and Biran teach all of the limitations of claim 27, they fail to teach recovering the original by combining the key with a hash value via a reverse function generator coupled to the memory wherein the reverse function generator restores the full key based on the one of the plurality of partial keys stored in the memory and the hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed, by applying the reverse hash function on the hash result and the hash identifiers for key restoration – col. 8,

lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

As for claim 31, Brandin teaches the hash value corresponds to a single entry in the memory (Fig. 1, each address contains pointer and confirmer information unique to the keys).

13. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Ji.

As for claim 14, Brandin fails to teach the apparatus of claim 7 further comprising a forwarding engine coupled to the memory, wherein the forwarding engine forwards a data packet according to information read from the memory at an address corresponding to the one of the plurality of partial keys stored in the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

Response to Arguments

14. Applicant's arguments with respect to rejections previously set forth under 35 U.S.C. §§ 102 and 103 have been fully considered but are moot in view of the new ground(s) of rejection.

15. Applicant's arguments with respect to the size of partial keys (being a majority of consecutive bits of the full key, requiring less storage, etc.) as presently recited in the newly amended claims (and newly added claims) as overcoming the rejections previously set forth, are rendered moot, as Examiner no longer relies on Fig. 13A in the rejection of these claims (i.e. wherein each full key contains two equally sized partial keys). Examiner maintains the claims in present form, are either anticipated or rendered obvious in part by Fig. 13B (and relevant portions of the specification related

to the figure). More specifically, Fig. 13B (element 324) depicts a full key (A+B+C) which contains 196 bits. The full key is split into a partial key (A+B), which contains 128 bits. The partial key itself (A+B) and the transform of this partial key (X(A) + X(AB)) require less storage space that the full key, and its respective transformation. Further explanation is provided in the rejections discussed *supra*.

Likewise Applicant's argument that Brandin fails to teach a hash value as **not** being stored in the memory is rendered moot, as Examiner maintains that Brandin in further view of Biran render this limitation obvious as per the rejections *supra*.

Applicant's general assertion that all dependant claims are allowable for at least further limiting a base claim with allowable subjection matter is rendered moot, as Examiner maintains that Brandin renders each claim either anticipated and/or obvious as per the rejections presented *supra*.

Conclusion

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Craig E. Walter whose telephone number is (571) 272-8154. The examiner can normally be reached on 8:30a - 5:00p M-F.

17. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung S. Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

18. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Craig E Walter
Examiner
Art Unit 2188

CEW

HYUNG SOUGH
SUPERVISORY PATENT EXAMINER
6-05-07